sitecore®

Sitecore CMS 6.0 or later

Workflow Reference

A Conceptual Overview for Architects and Developers

Table of Contents

Chapter 1

Introduction

This manual provides an overview of the workflow features for architects and developers.

This manual contains the following chapters:

- **Chapter 1 — Introduction**
  A brief introduction to this manual and its intended audience.

- **Chapter 2 — Understanding Sitecore Workflows**
  How workflows work from the perspective of the business user.

- **Chapter 3 — Defining and Configuring Workflows**
  How architects and developers can create, define, and configure workflows.

Chapter 2

Understanding Sitecore Workflows

Workflows ensure that items move through a predefined set of states before they become publishable, usually intended to ensure that content receives the appropriate reviews and approvals before publication to the live Web site.

This chapter describes workflows from the perspective of the business user.

This chapter contains the following sections:

- The Sample Workflow

- The Workbox

2.1 The Sample Workflow

Sitecore contains a sample workflow. The following description describes how the Sample Workflow works, which provides an illustration of how workflows work in general:

1. When a content author creates an item, the item is assigned to the *Draft* state. This occurs when the item's base template standard values item lists the *Sample Workflow* as its "Initial Workflow". The *Sample Workflow*, in turn, configures the *Draft* state as the workflow's "Initial State".

   The item cannot be published when it is in the *Draft* state, because the *Draft* state is not configured as a "final" state. Items in workflow states can only be published if the state they are in is configured as a final state.
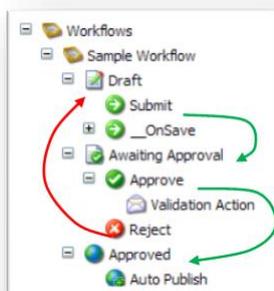
2. When the content author believes that the item is ready to be published, they click the *Submit* workflow command. The *Submit* workflow command is configured to move items to the *Awaiting Approval* state, which indicates to "reviewing users" that the content author believes that the content is ready for publication.

   The *Awaiting Approval* state has two workflow commands associated with it: *Approve* and *Reject*. The content author user, however, does not have execute access rights for these commands, so the commands do not appear in the user interface for the content author.

   While the item is in the *Awaiting Approval* state, however, the content author can make additional changes.

3. At some point, a user with the appropriate access rights will notice the item which is awaiting approval in the Sitecore Workbox. The reviewer can check the item content, including comparing it with previous versions if they exist, and decide whether the quality is good enough for the published Web site.

   The reviewer clicks the Reject command to send the item back to the *Draft* state and Sitecore prompts the reviewer for a comment to explain why the changes have been rejected.

   The reviewer clicks the *Approve* command to accept the changes.

4. When the reviewer clicks the *Approve* command, Sitecore triggers the *Validation* action, which checks that the item has no validation errors. This occurs because the *Approve* command includes a *Validation Action*. The validation action will cancel the movement of the item from the *Awaiting Approval* state to the *Approved* state if it finds errors.

   If the validation action finds no errors, the *Approve* command moves the item to the *Approved* state, which is configured as a "final" state.

5. When the item enters the *Approved* state, Sitecore triggers the *Auto-Publish* action configured in the state. This will automatically publish the item.

6. The next time a content author clicks Edit for this item to lock it before making changes, Sitecore automatically creates a new version of the item and places the new version in the *Draft* state, while the first version remains published.

Important
Make sure that content authors or other workflow users don't have a Sitecore administrator account. Otherwise, the workflow will behave differently. For more information about why the workflow behavior is different for a Sitecore administrator, see the section *Using a Workflow*.

2.2 The Workbox

The Workbox is an application in Sitecore that displays information about the items that are in a workflow and helps you monitor how items are moving through the workflow.

To open the Workbox, log in to the Sitecore Desktop and click **Sitecore**, **Workbox**.
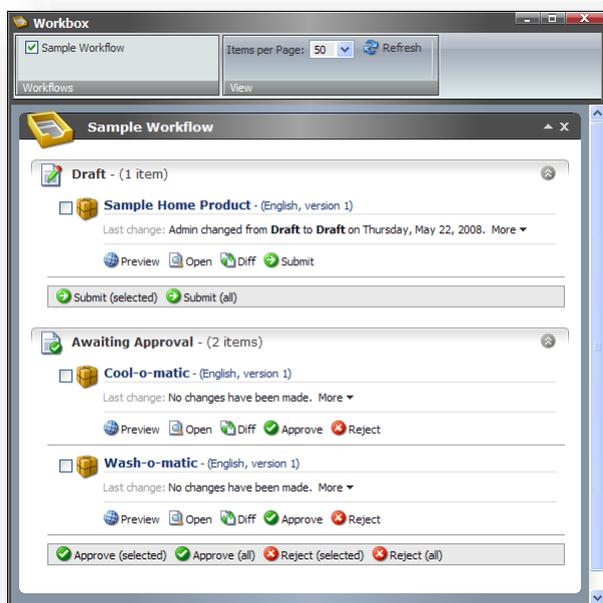
In the Workbox, you can see the editing history of each item in the workflow and the number of items in each state. You can select workflow commands associated with the workflow states if you have the Workflow Command Execute access right for the commands. Items in a workflow are grouped according to the state they are in.

The Workbox ribbon contains two groups — **Workflows** and **View**.

In the **Workflows** group, you can select the workflows that you want to view.

In the **View** group, you can specify the number of items per page.

The Workbox shows the following information for each item:
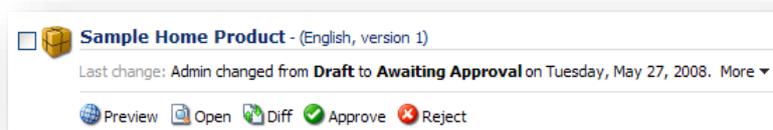
The name of the item is shown at the top of the item information. The checkbox to the left of the name is used for batch operations. The next line contains details of the last workflow change that was made

to the item. Click the **More** drop-down list to see the editing history of the item:
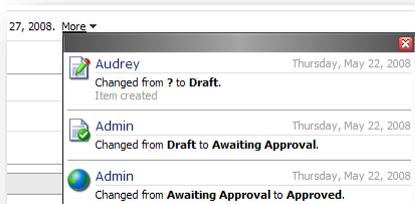
The bottom of the item information displays various commands. The first three commands are shown for all items. The remaining commands will vary depending on the workflow commands configured for the corresponding workflow state:

- Click **Preview** to see a preview of the current item.

- Click **Open** to open the item in the Content Editor.

- Click **Diff** to open the **Compare Versions** dialog box.

This dialog box displays the differences between two versions of the same item.

- Approve is a workflow command configured for this state. In this example, click Approve to accept the changes that have been made to the item and move it to the next stage in the workflow.

- Reject is a workflow command configured for this state. In this example, click Reject to reject the changes that have been made to the item and move it back to an earlier stage in the workflow.

The batch operations toolbar is located at the bottom of the **Workflow** window:

Note
Workflow states that do not contain any commands that the current user has execute access rights for are not shown in the Workbox.

Chapter 3

Defining and Configuring Workflows

Developers use the standard content items and fields stored in the system area of the content tree to define and configure workflows.

This chapter describes how to create and configure workflows.

This chapter contains the following sections:

- Defining Workflows

- Standard and Custom Workflow Actions

- Standard Template's Workflow Section Fields

- Workflow and Security

3.1 Defining Workflows

Workflows have a corresponding workflow definition item stored in the /*sitecore/System/Workflows* area of the content tree.

Workflow definition items contain one or more workflow state definition items configured as subitems. Workflow states define the steps a document must pass through before it becomes publishable.
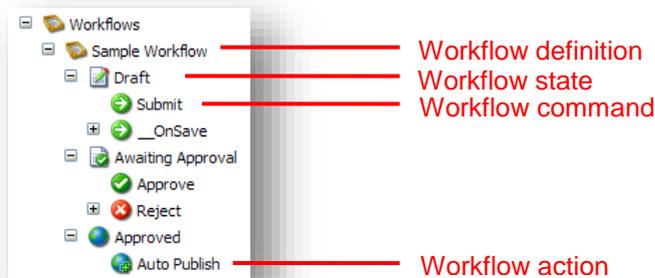
Workflow commands move items from one workflow state to another. Every workflow command has a corresponding definition item. The definition items for commands are stored under the corresponding workflow state definition item. The workflow command is only shown when the selected content item is in the corresponding workflow state.

You can associate workflow actions with either workflow states or workflow commands. To do so, you must create a workflow action definition item as a subitem under the appropriate workflow state or command. The workflow action is enabled when an item enters the corresponding workflow state. The workflow action associated with a command is performed when a user selects the command.

An action definition item indicates a method that Sitecore calls when the action is raised. The method can do anything appropriate, such as send an e-mail message.

3.1.1 The Workflow Definition Item

Workflow definition items are stored under /*sitecore/System/Workflows*. The workflow definition items define workflows and have workflow state definition items as child items.

Workflow definition items contain the **Initial State** field. The **Initial State** field defines the state that a new item is assigned when it is part of this workflow.
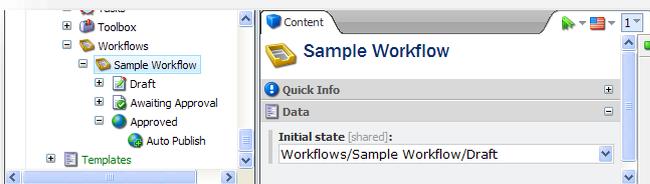
Workflow definition items are based on the /*sitecore/templates/System/Workflow/Workflow* template.

3.1.2 Workflow State Definition Item

Workflow state definition items can have workflow command definition items or workflow action definition items as their child items.
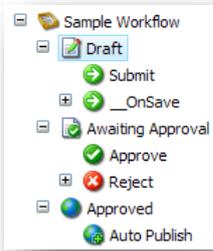
The actions attached to a workflow state are executed automatically when an item reaches this state.

Workflow commands are used to move items from one workflow state to another. Commands can also trigger workflow actions.

Workflow state definition items are based on the */sitecore/templates/System/Workflow/State* template.

The workflow state template contains the **Final** checkbox. If the **Final** checkbox is selected, then any items in this state are publishable. When a user clicks Edit to lock the item that is in a final workflow state, Sitecore automatically creates a new version of the item, checks out this version, and places it in the initial workflow state.

Hint
Deleting the associated version will undo any editing that has been done.

Items do not necessarily need to flow through all the workflow states. Custom commands or actions can send items directly to any workflow state.

3.1.3 Workflow Command Definition Item

Workflow commands move items between workflow states and to trigger workflow actions. In the Content Editor, the commands that are attached to a workflow state are shown on the **Review** tab in the **Workflow** group when an item is in this state and the current user has access to these commands. The same commands are also shown in the Workbox.
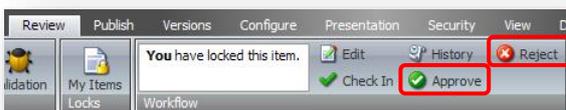
Workflow command definition items are based on the */sitecore/templates/System/Workflow/Command* template.

The *Command* template contains the following fields:

| Field	Value
Next State	The state that the item will be moved to after the action is executed.
Suppress Comment	This checkbox defines whether or not users are prompted to enter a comment when a workflow command is executed. If the checkbox is cleared, users are prompted. If the checkbox is selected, users are not prompted.

__OnSave Command

This command is a placeholder for the actions that should take place when a user saves changes to an item in this workflow state. This functionality is defined in the `saveUI` pipeline in the `web.config` file.

In the *Sample Workflow,* the _OnSave command triggers the *Auto Submit Action*. When users who are members of the *Sitecore Minimal Page Editor* role save an item, the *Auto Submit Action* automatically moves the item to another workflow state. This is done because members of the Sitecore Minimal Page Editor role do not have access to the workflow related commands in the Page Editor.
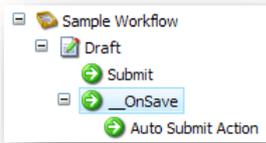
3.1.4   Workflow Action Definition Item

Workflow action definition items can be created under a workflow state item or under a workflow command item.
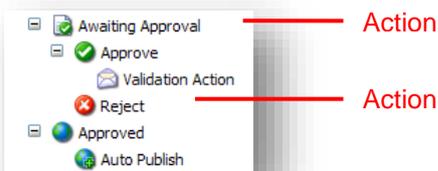
If a workflow action definition item is created under a workflow state item, the action is performed when an item enters the workflow state.

If a workflow action definition item is created under a workflow command item, the action is executed when the command is triggered.

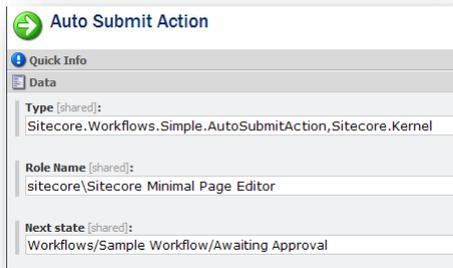Use the */sitecore/templates/System/Workflow/Action* template to create custom actions.

Sitecore also contains some predefined workflow actions. These are described in the following section.

3.2 Standard and Custom Workflow Actions

Sitecore contains a number of standard workflow actions. You can also create a custom workflow action by creating an action definition item and configuring it to call a .NET method that you must also create.

3.2.1 Auto Submit Action

When triggered, the *Auto Submit Action* automatically moves the current item to another state if the current user belongs to a specific role.

This action is based on the */sitecore/templates/System/Workflow/Auto Submit Action* template.

The **Data** section of this workflow action contains the following fields:

Field	Value
Type	The construct "namespace.classname, assembly_name". The specified class defines the functionality of the action.
Role Name	The name of the role that this action is available to. The action will not be available to all the other roles. This field should not be empty (if it is empty, you receive an error). You can only enter one role.
Next State	The next state. When the action is triggered, it moves the current item to the state that is specified in this field.

3.2.2 Auto Publish Action
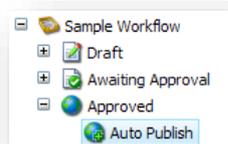
This action appears under the Approved state of the Sample Workflow.

It is based on the common */sitecore/templates/System/Workflow/Action* template.

The data section of this workflow action contains the following fields:

Field	Value
Type	The namespace.class, assembly name of the implementation class. For example: Sitecore.Workflows.Simple.PublishAction, Sitecore.Kernel
Parameters	The deep parameter that specifies whether or not the child items should be published. deep=1 — publish children. deep=0 — do not publish children.

3.2.3   Email Action

Use this action to send e-mail messages.

This action is based on the */sitecore/templates/System/Workflow/Email Action* template.

This template contains the following fields:

Field	Value
To	The e-mail address of the recipient
From	The e-mail address of the sender
Subject	The subject of the e-mail
Message	The message. The default field type is Memo. You can change this field to send messages of any type.
Mail server	The SMTP server used to send e-mails.
Type	Use this field to override the provided .NET implementation method with a custom method.

3.2.4   Validation Action

Use this action to execute the validation rules specified in **Workflow Validation Rules** field of an item.

The action is based on the */sitecore/templates/System/Workflow/Validation Action* template.

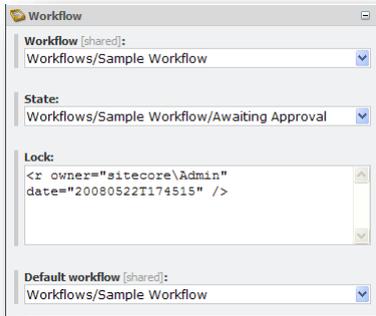This template contains the following fields:

Field	Value
Type	This field contains the following default value: Sitecore.Workflows.Simple.ValidatorsAction, Sitecore.Kernel. This field is inherited from the default Action template.

Field	Value
Max Result	The maximum response from the validator. The possible values are: • Unknown • Valid • Suggestion • Warning • Error • CriticalError • FatalError If the value of this field is "Error", then items which have errors will pass, but the items which have critical errors will not pass. Errors which result in specific validator responses are defined by the validator classes.
Unknown	The message shown to the user when the validation result is Unknown.
Warning	The message shown to the user when the validation result is Warning.
Error	The message shown to the user when the validation result is Error.
Critical Error	The message shown to the user when the validation result is Critical Error.
Fatal Error	The message shown to the user when the validation result is Fatal Error.

3.3 Standard Template's Workflow Section Fields

The Standard Template inherits the Workflow template.

This template contains the following fields:

Field	Value
Workflow	The workflow currently attached to the item.
State	The current workflow state.
Lock	Whether or not the item is locked by a user. If the item is locked, this field shows the user and the time when the item was locked.
Default Workflow	The default workflow of the item.

3.3.1 Assigning Workflows to Items

By default, items are not placed in a workflow when created. Sitecore only places items in a workflow if the "Initial" workflow is set for the corresponding data template standard values item. The Initial workflow corresponds to the **Default Workflow** field in the Standard Template.

3.4 Workflow and Security

Sitecore defines three workflow specific access rights.

- **Workflow State Delete** — controls whether or not a user can delete items which are currently associated with a specific workflow state

- **Workflow State Write** — controls whether or not a user can update items which are currently associated with a specific workflow state.

- **Workflow Command Execute** — controls whether or not a user is shown specific workflow commands.

The access rights that you have to a content item can influence the behavior of the Workbox application. You must have write access to an item in order to see the item in the Workbox. You may not have write access to an item if it is currently checked out (locked) by another user.

3.4.1 Creating and Editing a Workflow

To create or edit a workflow, you must:

- Unprotect the *Workflows* item — `/sitecore/system/workflows`.

- Have *READ* and *CREATE* permissions for the *Workflows* item.

- Be a member of the *Sitecore Client Authoring* role.

To assign permissions to workflow states or commands, you must be a member of the *Sitecore Client Securing* role.

3.4.2 Using a Workflow

A workflow that has been created by a Sitecore administrator, who is a member of the *Sitecore Client Authoring* role, is used by different types of users, such as web administrators or content editors, who must have basic content authoring rights or read access to the workflow.

For workflow user accounts, Sitecore creates a new version every time an item in a final state (for example, *Approve*) is edited and moves it to the initial state. For administrator accounts, Sitecore doesn't create a new version and doesn't move it to the initial state.

Important
Make sure that workflow users don't have an administrator account and that they are not members of the *Sitecore Client Authoring* role and the *Sitecore Client Securing* role.

3.4.3 Hiding a Workflow State from Certain Users

Users who have read access to a workflow state can see that state in their workbox as long as the state includes workflow commands for which they have command execute access rights. If business requirements state that a particular workflow state should be hidden from a given set of users, you can restrict access to that state for those users by:

- Hiding all the workflow commands in the state from the users in question.

  or

- Explicitly hiding the workflow state itself from the users in question.

To explicitly hide a workflow state:

- Turn off the inheritance access right for the workflow state item and do not grant read access to the workflow state to the user and all the roles assigned to the user.

  or

- Deny the user or one of the roles that the user is assigned read access to the workflow state item.

Each of these approaches has its advantages and disadvantages.

- Turning off the inheritance access right means that you must explicitly grant access to all the roles that should be able to see the workflow state in the Workbox. This is the best approach when only a small number of users and roles need to see the workflow state in the Workbox.

- In the Sitecore security system, deny always overrules allow. When you explicitly deny a role read access, you can inadvertently prevent a user who has been assigned many roles from seeing the workflow item. Denying read access can have unanticipated results.

In general, we recommend that you turn off the inheritance access right and explicitly allowing access rights when the number of roles that require access is manageable.

3.4.4   Hiding a Workflow Command for Certain Users

The Content Editor and Workbox only displays workflow commands for non-Administrator users when:

- The user has write access to the associated item or item is locked by the user.

  and

- The user has "Workflow State Write" access to the command's parent workflow state or item is locked by the user.

  and

- The user has read access to the workflow command itself.

  and

- The user has the "Workflow Command Execute" access rights to the command itself.

If you configure the Sitecore security settings so that a user does not meet one of these criteria, you will hide the workflow command from that user.

If the user must have write access to both the item and the workflow state, there are two ways to deny them read access to the workflow command.

- Turn off the inheritance access right for the workflow command item and do not grant read access to the workflow command to the user and all the roles that the user is a member of.

- Deny read access for the workflow command item to the user or one of the roles that the user is a member of.

Each of these approaches has its advantages and disadvantages.

- Turning off the inheritance access right means that you must explicitly grant access to all the roles that should be able to see the workflow state in the Workbox. This is the best approach when only a small number of users and roles need to see the workflow state in the Workbox.

- In the Sitecore security system, deny always overrules allow. When you explicitly deny a role read access, you can inadvertently prevent a user who has been assigned many roles from seeing the workflow item. Denying read access can have unanticipated results.

In general, we recommend that you turn off the inheritance access right and explicitly allowing access rights when the number of roles that require access is manageable.